

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

REMARKS/ARGUMENTS

By this Amendment, the Abstract is amended and Claims 1-20 are amended. Claims 1-20 are pending.

Favorable reconsideration is respectfully requested in view of the foregoing amendments and the following remarks.

The Examiner sets forth that the abstract of the disclosure is objected by the Examiner to because the Examiner believes that it exceeds the limit of 150 words. Correction is required by the Examiner. The Examiner directs the Applicants' attention to MPEP § 608.01(b).

The Abstract has been amended accordingly.

The Examiner further sets forth that Claims 2-5, 7-13 and 15-20 are objected by the Examiner because of the following informalities:

(i) Dependent Claims 2-5 should begin with "the method" as they are referred to "a method" of independent Claim 1.

(ii) Dependent Claims 7-13 should begin with "the method" as they are referred to "a method" of independent Claim 6.

(iii) Dependent Claims 15-20 should begin with "the execution system" as they are referred to "an execution system" of independent Claim 14.

Appropriate correction is required by the Examiner.

The claims have been amended accordingly.

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

The Examiner further sets forth that Claims 1-20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The Examiner believes that the following phrases lack antecedent basis:

- (i) "the stack frame template" (Claim 1, lines 4-5 and Claim 6, lines 6-7)
- (ii) "the method's exact stack requirements" (Claim 1, lines 6-7 and Claim 6, lines 8-9)
- (iii) "the execution speed" (Claim 14, line 1)

The Claims have been amended accordingly.

The Examiner sets forth that Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Alexander III, et al. (U.S. 6,507,946) in view of Brandy et al. (U.S. 6,654,778).

As to Claim 14, the Examiner believes that Alexander teaches this invention substantially as claimed including an execution system for increasing the execution speed of invoking Methods of one or more classes (the Examiner directs the Applicants' attention to abstract, lines 1-2 and col. 1, lines 9-10), the system comprising:

memory (the Examiner directs the Applicants' attention to 424; col. 5, line 35 and Fig. 4); and

a virtual machine (the Examiner directs the Applicants' attention to 400; col. 5, lines 27-30 and Fig. 4) operable to access the memory, to create a representation of at least one of the Methods based on an activation stack frame template with a set of criteria (the

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

Examiner directs the Applicants' attention to col. 5, lines 46-57), to create a representation of at least one of the Methods based on exact stack requirements (the Examiner directs the Applicants' attention to col. 5, lines 57-60), and to spatially associate a Method access structure contiguous to the representation of each of the Methods (the Examiner directs the Applicants' attention to col. 6, lines 13-19).

According to the Examiner, Alexander does teach a Java virtual machine 400 including Execution engine 406 for executing instructions contained in the methods of classes loaded by class loader subsystem 402 (col. 5, lines 29-38), but according to the Examiner does not explicitly teach creating a Method routing structure for each of the one or more classes.

The Examiner further sets forth that Blandy teaches creating a Method routing structure for each of the one or more classes (the Examiner directs the Applicant's attention to col. 6, lines 9-14).

According to the Examiner it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Blandy with Alexander because the Examiner believes that Blandy's teachings would have provided the capability for insuring that the parameters are passed appropriately to the target method, avoiding function activation and interpretation overhead for calls to selected methods in the Java virtual machine interpreter. Therefore, program execution speed is greatly improved.

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

As to Claim 15, the Examiner believes that Alexander teaches the set of criteria includes the number of parameter words, the total number of local words, and the number of words of evaluation stack (the Examiner directs the Applicant's attention to Fig. 5B and associated text).

As to Claim 16, the Examiner believes that Alexander teaches the activation frame template includes a local variable portion (the Examiner directs the Applicant's attention to col. 5, line 53), an evaluation stack (the Examiner directs the Applicant's attention to col. 5, lines 55-57), and a fixed size frame linkage structure (the Examiner directs the Applicant's attention to col. 6, lines 55-60).

As to Claim 17, the Examiner believes that Alexander does teach a Java virtual machine 400 including Execution engine 406 for executing instructions contained in the methods of classes loaded by class loader subsystem 402 (the Examiner directs the Applicant's attention to col. 5, lines 29-38), but does not explicitly teach associating a pointer with each Method access structure, the pointer defined such that it is an indicator of where code for implementing a Method resides and an indicator for the Method itself.

The Examiner sets forth that Blandy teaches associating a pointer with each Method access structure, the pointer defined such that it is an indicator of where code for implementing a Method resides and an indicator for the Method itself (the Examiner directs the Applicant's attention to col. 6, lines 24-31).

According to the Examiner it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Blandy with

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

Alexander because the Examiner believes that Blandy's teachings would have provided the capability for insuring that the parameters are passed appropriately to the target method.

As to Claim 18, the Examiner believes that Alexander teaches the virtual machine is operable to maintain Method access structures associated with dynamically compiled code in an area of memory separate from Method access structures associated with bytecode (the Examiner directs the Applicant's attention to col. 5, lines 16-26).

As to Claim 19, the Examiner sets forth that Alexander does not explicitly teach that the Method routing structure includes one or more misaligned pointers to denote processor executable Method access structures.

According to the Examiner, Blandy teaches the Method routing structure includes one or more misaligned pointers to denote processor executable Method access structures (the Examiner directs the Applicant's attention to col. 6, lines 9-14).

Thus, the Examiner sets forth that it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Blandy with Alexander because the Examiner believes that Blandy's teachings would have provided the capability for insuring that the parameters are passed appropriately to the target method.

As to Claim 20, the Examiner sets forth that Alexander teaches the virtual machine is operable to spatially associate the Method access structure immediately preceding the

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

representation of each of the Methods (the Examiner directs the Applicant's attention to col. 6, lines 13-19).

As to Claim 1, the Examiner sets forth that Alexander teaches the invention substantially as claimed including a method of increasing the execution speed of invoking and returning from a Method while reducing the supporting memory footprint (the Examiner directs the Applicant's attention to abstract, lines 1-2 and col. 1, lines 9-10), the method comprising:

establishing an activation stack frame template with a set of criteria (the Examiner directs the Applicant's attention to col. 5, lines 47-55);

determining whether the Method conforms to the criteria of the stack frame template (the Examiner directs the Applicant's attention to col. 7, lines 23-32);

creating a fixed size activation frame regardless of the Method's exact stack requirements, based on the set of criteria of the activation stack frame template if the Method conforms to the set of criteria of the activation stack frame template (the Examiner directs the Applicant's attention to Fig. 5B and associated text); and

associating a Method access structure with the Method such that the Method access structure is contiguous with the code of the Method (the Examiner directs the Applicant's attention to col. 6, lines 13-19).

The Examiner sets forth that Alexander does teach an activation frame (the Examiner directs the Applicant's attention to col. 6, lines 32-37) and the activation stack frame template, but does not explicitly teach an activation frame to match the Method's

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

exact stack requirements if the Method does not conform to the set of criteria of the activation stack frame template.

According to the Examiner, Blandy teaches an activation frame to match the Method's exact stack requirements if the Method does not conform to the set of criteria of the activation stack frame template (the Examiner directs the Applicant's attention to col. 7, lines 21-51).

The Examiner further sets forth that it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Blandy with Alexander because the Examiner believes that Blandy's teachings would have provided the capability for avoiding function activation and interpretation overhead for calls to selected methods in the Java virtual machine interpreter. Therefore, program execution speed is greatly improved according to the Examiner.

As to Claims 2-4, the Examiner directs the Applicant's attention to the rejection of Claims 15-17 above. The Examiner sets forth that Claims 2-4 are the same as Claims 15-17, except Claims 2-4 are method claims and Claims 15-17 are system claims.

As to Claim 5, according to the Examiner Alexander teaches the Method access structure is variably sized (the Examiner directs the Applicant's attention to Fig. 5A and to the associated text).

As to Claim 6, the Examiner sets forth that the rejection of Claim 1 above is incorporated herein in full. Alexander, however, does not explicitly teach creating a Method

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

routing structure for each class according to the Examiner; and rewriting invocation bytecodes to a form that includes an indication of the Method routing structure.

The Examiner further sets forth that Blandy teaches creating a Method routing structure for each class (the Examiner directs the Applicant's attention to col. 6, lines 9-14); and rewriting invocation bytecodes to a form that includes an indication of the Method routing structure (the Examiner directs the Applicant's attention to col. 8, lines 16-18).

The Examiner believes that it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Blandy with Alexander because the Examiner believes that Blandy's teachings would have provided the capability for avoiding the overhead involved with function activation and interpretation of the methods. Therefore, increasing performance in execution of methods according to the Examiner.

As to Claims 7-11, the Examiner directs the Applicant's attention to the rejection of Claims 15-19 above. The Examiner sets forth that Claims 7-11 are the same as Claims 15-19, except Claims 7-10 are method claims and Claims 15-19 are system claims.

As to Claim 12, the Examiner directs the Applicant's attention to the discussion of Claim 19 above for rejection of "the misaligned pointers are used to denote processor executable Method access structures." Additionally, the Examiner believes that Blandy further teaches one or more aligned pointers are used to denote processor non-executable Method access structures (the Examiner directs the Applicant's attention to col. 6, lines 25-28).

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

The Examiner believes that it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Blandy with Alexander because the Examiner believes that Blandy's teachings would have provided the capability for increasing performance in execution of methods by avoiding the overhead involved with function activation and interpretation of methods.

As to Claim 13, the Examiner directs the Applicant's attention to Claim 5 for rejection.

The Applicant's Invention

The Applicant's invention is a system for increasing the execution speed of invoking Methods and returning from Methods. In the Applicant's system, a template having a set of predetermined criteria is developed and each Method from the subject code is compared to the template in order to determine whether it conforms to the criteria. The same template is compared with each of the different Methods to determine the size of its activation frame. Different methods for creating activation frames of different sizes are used in the Applicant's system depending of the results of the comparison with the template. Thus, the Applicant's criteria are frame size determining criteria and the Applicant's template is a frame size determining template.

Each Method from the Applicant's subject code is thus examined to determine whether it conforms to the criteria of the frame size determining template. If the Method conforms to the template criteria, an activation frame for that Method is created based on the template when the Method is invoked. Otherwise, an activation frame based on

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

standard virtual machine criteria is created when the Method is invoked, thereby providing a stack frame of the exact size.

In a preferred embodiment of the invention, the Methods are spatially optimized to provide Method access structures. The Method access structures are then associated with the Method such that a Method access structure and its corresponding Method are contiguous with each other. Additionally, a Method routing structure is provided to support all the Method access structures of a class. In order to facilitate execution by the system, the Method routing structures, which are external to the Method access structures and pointing to the Method access structures, are provided with pointers to point execution to the executable Method access structures.

The size determining criteria of the size determining template can include a count of the number of words required by a Method. For example, the criteria can be that a Method require no more than a predetermined number of parameter words, local words, evaluation words, etc. In the case where the number of words required by a Method is compared in this manner, the determination of the activation stack size is therefore a determination of a word number requirement.

Therefore, the Applicant's amended Claim 1 sets forth a method of increasing the execution speed of invoking and returning from a Method of a plurality of Methods executing on a computer system while reducing the supporting memory footprint, the method including the steps of establishing a single frame size determining template for comparing the single frame size determining template with each of the Methods of the

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

plurality of Methods, the single frame size determining template having a set of frame size determining criteria representative of a predetermined number of words for determining the size of activation frames when creating the activation frames. The Applicant's method also includes determining whether a word number requirement of the Method conforms to the frame size determining criteria of the single frame size determining template and conditionally creating a predetermined fixed size activation frame regardless of an exact stack requirement of the Method, based on the set of frame size determining criteria of the single frame size determining template if the word number requirement of the Method conforms to the set of frame size determining criteria of the single frame size determining template. Conditionally creating an activation frame to match the Method's exact stack requirements if the word number requirement of the Method does not conform to the set of frame size determining criteria of the single frame size determining template is also recited. Also included are spatially optimizing the Method to provide a Method access structure and associating the Method access structure with the Method such that the Method access structure is contiguous with the code of the Method.

The Applicant's amended Claim 6 sets forth a method of increasing the execution speed of invoking a plurality of Methods in an execution device, the plurality of Methods associated with one or more classes including the steps of establishing a single frame size determining template for comparing the single frame size determining template with each of the Methods of the plurality of Methods when they are invoked, the single frame size determining template having a set of frame size determining criteria representative of a

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

predetermined number of words for determining the size of activation frames when creating the activation frames and determining whether a word number requirement of the one Method conforms to the frame size determining criteria of the single frame size determining template and conditionally creating a fixed size activation frame regardless of an exact stack requirement of the one Method, based on the set of frame size determining criteria of the single frame size determining template if a word number requirement of the one Method conforms to the set of frame size determining criteria of the single frame size determining template. The Applicant's method also includes conditionally creating an activation frame to match the one Method's exact stack requirements if the word number requirement of the one Method does not conform to the set of frame size determining criteria of the single frame size determining template. Spatially optimizing the Method to provide a Method access structure and associating the Method access structure with the Method such that the Method access structure is contiguous with the code of the Method are also included. A Method routing structure external to the Method access structure is created for each class and invocation bytecodes are rewritten to a form that includes an indication of the Method routing structure.

The Applicant's amended Claim 14 sets forth an execution system for increasing an execution speed of invoking Methods of one or more classes, the system including memory and a virtual machine operable to access the memory, to create a representation of at least one of the Methods based on a single frame size determining template for comparing the single frame size determining template with each of the Methods of the

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

plurality of Methods when they are invoked. In the Applicant's system the single frame size determining template has a set of frame size determining criteria representative of a predetermined number of words to determine the size of activation frames when creating the activation frames and to conditionally create a representation of at least one of the Methods based on exact stack requirements. At least one of the Methods is spatially optimized to provide a Method access structure and to dispose the Method access structure contiguous to the representation of each of the Methods. A Method routing structure external to the Method access structure and pointing to the Method access structure is created for each of the one or more classes in the memory.

References Cited by Examiner

Alexander teaches a system for optimizing an invocation of a Method. In the Alexander system, a determination is made to compile a calling Method and a call to a callee Method is detected within the calling Method. The callee Method is then inlined with the calling Method. When the new thread is launched for the called Method, the JVM creates a new stack for the thread. The stacks are composed of stack frames for storing the state of the local variables of the thread.

However, the Alexander reference is silent with respect to using a frame size determining template having frame size determining criteria for determining the size of the frames created in this manner as required in the Applicant's claimed invention. Very significantly, Alexander teaches the comparison of a plurality of templates of differing sizes

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

with the Methods during its verification checks, rather than a single template for all of the templates as also required in the Applicant's claimed invention.

Alexander teaches performing several verification checks in connection with the foregoing if symbolic references must be resolved. These checks insure that Java language rules are followed, that the invoke instructions can be safely executed and that the current class can legally access the Method. If any of the checks fail, an exception is thrown. However, there is no teaching in Alexander to determine, in particular, whether a word number requirement is met in order to proceed as required in the Applicant's claimed invention.

Furthermore, it is significant that the Alexander system does not proceed to provide frames of differing sizes in accordance with the foregoing verification checks. Rather, as described, it proceeds either to create a stack frame or to throw an exception. Thus, the verification checks of Alexander are not frame size determinations as required in the Applicant's claimed invention.

Alexander does teach creating stack frames and creating different kinds of stack frames depending on whether an instance Method is of type "reference" or "int.". However, there is no teaching of providing differing sizes of stack frames depending on a comparison between the stack requirements and a size determining template wherein the comparison is based upon a word number requirement as required in the Applicant's claimed invention.

Furthermore, Alexander does not teach spatially optimizing a Method and disposing the resulting Method access structure contiguous with the code of the Method. The

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

Alexander reference does teach placing class-type information in the Method area. However, the teaching merely that class-type information is placed into an area of memory does not suggest placing a Method access structure in the area, since the Applicant's Method access structure embodies representations of the bytecodes in the Method rather than a class-type. Furthermore, teaching the placement of two code fragments merely in the same area of memory does not suggest making them contiguous with each other as required in the Applicant's claimed invention.

Blandy teaches a Method for avoiding function activation and overhead in a Java environment. The use of templates is taught in Blandy. However, the templates taught by Blandy are used to determine whether an invoker should be called. Thus, the Blandy template is not a frame size determining template for determining the size of stack frames to be created as required in the Applicant's claimed invention. It follows that there are no frame size determining criteria representative of a number of words or determinations of word number requirements.

The Examiner cites Blandy as teaching creation of a Method routing structure for each one of the one or more classes. Blandy does teach an invoker pointer. However, the invoker pointer taught by Blandy resides within the Method rather than external to the Method as required for the Method routing structure of the Applicant's claimed invention. Furthermore, the pointer of Blandy invoker points to an invoker rather than to an executable Method access structure as required in the Applicant's invention. Since Blandy

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

does not teach a Method routing structure it follows that it does not teach rewriting invocation bytecodes to include an indication of a Method routing structure.

The Examiner also sets forth that Blandy teaches an activation frame to match the Methods exact stacked requirements if the Method does not conform to the set of criteria of the activation stack frame template. However, the Applicants respectfully submit that even though the Blandy does teach creating an activation frame to match the exact stack requirements of the Method, it does not create an activation frame conditionally dependent upon a comparison with a frame size determining template having a set of frame size determining criteria representative of a predetermined number of words as required in the Applicant's claimed invention.

Furthermore, Blandy does not teach spatially optimizing the Method and disposing the resulting Method access structure contiguous with the code of the Method as required in the Applicant's claimed invention.

Claims 2-5 are dependent from Claim 1. Claims 7-13 depend from Claim 6. Claims 15-20 depend from Claim 14. Therefore, for the reasons described above, these claims also allowable.

For at least the reasons set forth above, it is respectfully submitted that the above-identified application is in condition for allowance. Favorable reconsideration and prompt allowance of the claims are respectfully requested.

Application No. 09/996,169
Amendment Dated January 18, 2005
Reply to Office Action of September 21, 2004

Should the Examiner believe that anything further is desirable in order to place the application in even better condition for allowance, the Examiner is invited to contact Applicant's undersigned attorney at the telephone number listed below.

Respectfully submitted,

CAESAR, RIVISE, BERNSTEIN,
COHEN & POKOTILOW, LTD.

January 18, 2005

Please charge or credit our Account
No. 03-0075 as necessary to effect
entry and/or ensure consideration of
this submission.

By Frank Linguiti
Frank M. Linguiti
Registration No. 32,424
Customer No. 03000
(215) 567-2010
Attorneys for Applicant